Solving fake news challenge task based on ensemble learning

Desiatkin Dmitrii, Rassabin Maksim
Innopolis University
d.desyatkin@innopolis.univercity,m.rassabin@innopolis.ru

Abstract

The task of stance detection for body-title pairs takes an important place in throughout the fake news detection process. As the results showed Fake news challenge - 1 the models with ensemble learning take bests places in the leaderboard. The presented work was considered an ensemble learning approach for stance detection. the main types of feature decoders for the stance detection task: keywords features based on manual selection and mutual information approach, words factorization, word to vector decoder. Leaning on challenge result a simple neural network chosen as the baseline for ensemble members and single analyze. Each feature type was analyzed with the main challenge's metrics and based on these results created the ensemble of 3 models, which bit the top approach for Fake news Challenge. The project development vector and ways to improve it based on modern approaches to the presented task were also noted.

Keywords: stance detection, fake news,nature language processing,deep learning,ensemble learning,word's feature

1. Introduction

In recent years, the problem of determining false information has been particularly acute. Thanks to modern technology and research in the field of neural networks, it has become possible to create such types of artificial content like photos, videos, music, voice simulations. Also, thanks to the research of Open AI team and gpt2 model, it became possible to generate a high-quality text. However, these technologies also allow the creation of high-quality false content, which makes it increasingly difficult to identify. Fake news challenge considers one of the subtasks for identifying false textual information, namely stance detection. Stance Detection involves estimating the relative perspective (or stance) of two pieces of text relative to a topic, claim or issue. FNC-1 was chosen as the task of estimating the stance of a body text from a news article relative to a headline. Specifically, the body text may agree, disagree, discuss or be unrelated to the headline. In order to solve this problem, the approach that was described in github repository was considered. In this project, the author implemented deep ensemble learning based on 3 top approaches from FNC - 1. Important to mention that he has not taken top models and (Liao, 2018)

2. Background

Since a special interest in information verification has arisen since 2016, studies on the correspondence of the title and content of articles appeared even before of the FNC. (Chen, 2017; Rakholia, 2017). This approach was implemented as LSTM-based RNN models, but insofar as models do not get good enough results it's not presented on the official competition.

The fake news challenge project lasted less than a year in 2017, so there is not much research on this specific task. However, the most similar research area is Natural Language Inference (NLI), which is defined as stance detection in tweets and online debates are quite common now. In this direction, there are some modern papers using domain adaptation (Xu et al., 2019), transfer learning (Rao, 2019) and Deep Bidirectional Transformers (Fajcik et al., 2019). But, there is a significant difference between the two tasks: the latter concerns determining the position in a statement in a natural language instead of a specific goal.

According to official data, 50 participants took part in the competition, the leader board of which is shown on Figure 1.

Rank	Team name	Affliation	Score	Relative Score
1	SOLAT in the SWEN	Talos Intelligence	9556.50	82.02
2	Athene (UKP Lab)	TU Darmstadt	9550.75	81.97
3	UCL Machine Reading	UCL	9521.50	81.72

Figure 1: Top 3 of FNC-1

The Talos Intelligence team won this challenge also using the ensemble prediction (Cisco-Talos, 2018), where the main scheme on Figure 2. The first ensemble model is a convolution neural network model and the second one is a gradient boosted decision trees with five overarching sets of features as inputs.

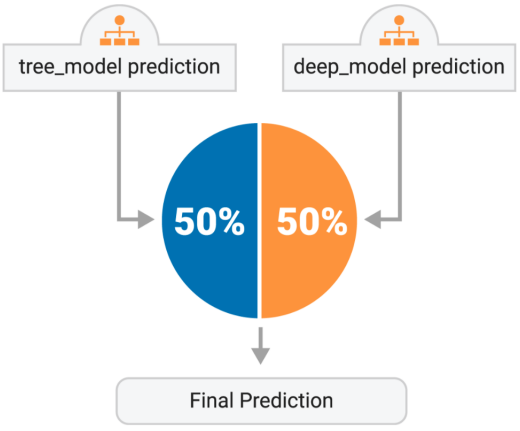

Figure 2: Prediction scheme of Talos Intelligence team

The Athene team consider ensemble voting based on 5 models with different extracted features (Hanselowski,):

- Baseline features from the FNC-1
- Non negative matrix factorization cosine distance
- Non negative matrix factorization concatenated

- Bag of words unigram features
- Latent Dirichlet Allocation

Each model based on multi layer perceptron.(Uclnlp, 2018) UCL team used bag-of-words representations for the text inputs in neural networks: term frequency (TF) and term frequency-inverse document frequency (TF-IDF). UCL mentioned that used simple structure which allowed to get similar result with more complex ensemble models.

3. Methodology

3.1. Baseline

As mentioned authors of challenge baseline for overall challenge can be considered like hand-coded features with simple a GradientBoosting classifier. With these features and a gradient boosting classifier, the baseline achieves a weighted accuracy score of 79.53% with a 10-fold cross validation.. As baseline approach for work was chosen UCL's model in terms of simplicity, computational complexity and competitive results. The basic frame is an MPL with a hidden layer of 100 units. The output layer is a four-dimension vector representing the raw predictions that classification model generates. In baseline model for feature input was chosen feature extracted by Term Frequency times Inverse Document Frequency approach (Leskovec et al., 2016). This method was applied independent to article's text, to the body's text and get 2 feature vectors. And based on it was created 3'rd feature vector as cosine difference between them.
The loss function was defined as a summation of two parts. Part one is the cross entropy between the system's softmax probabilities and the true labels. Part two is the l2 regularization of the MLP weights.

3.2. Dataset

The training dataset consists of two parts. The first part is a file containing 1683 article bodies, each with a unique body ID. The second part is a file containing 49972 pairs of article headline and body ID.

rows	unrelated	discuss	agree	disagree
49972	73.13%	17.82%	7.36%	1.68%

Table 1: Composition of dataset.

Each pair is labelled with a stance in the domain of agree, disagree, discuss, unrelated, which is the body text's stance with regard to the body's headline. Among all the pairs, if removing duplicated headlines, only 1643 unique headlines exist, hence it is a many to many mappings between headlines and bodies. The distribution of stances is shown in Table 1.

3.3. Evaluation metric

FNC-1 proposed a hierarchical evaluation metric as illustrated in Figure 3.It involves two steps: 1) Correctly classify headline/body pair as related or unrelated contributes 0.25 points; 2) Correctly classify related pairs as agree, disagree or discuss contributes 0.75 points. This score weighting schema is designed for the consideration that identifying the relating stance is easier than discovering a stance towards an orientation. This metric is considered as the evaluation method for the FNC-1 competition.

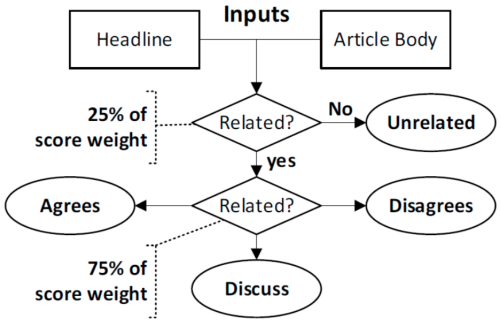

Figure 3: Evaluation scheme

Besides, another evaluation metric, macro-averaged F_1 scores (F_1m is proposed for overall model performance evaluation). To calculate the F_1m score, the class-wise F_1 cores, which are the harmonic means of the precision and recalls of the four classes, are calculated first. Then these class-wise F_1cores are averaged to generate F_1m. F_1m is more appropriate for validating the document-level stance detection task because it does not have the bias preference for models performing well on the majority class and poorly on the minority class.

3.4. Feature vector selection

In considered work author decide to find the optimal sequence of the ensemble. For each model, the same NN graph was used. The only difference is the feature vectors that were given as input of the model. We can split the models into 3 groups by feature vectors type.

The first type of features has been described in the baseline part.

The second type of feature vector is based on keywords extracted from the overall dataset. As shown on FCN-1 results the worst classified classes are 'agree' and 'disagree'. Therefore extracting keywords was performed in order to increase the accuracy of those classes. Based on the refuting/key words indicator vector was constructed which is indicating the presence of such words in the input document. This is our new feature vector.
The simplest model to consider for this feature type is the manual extraction of keywords. A list of discussion words was manually selected according to error analysis and vectorized to be concatenated to the input vector.
In Spite of good results, the manual method not robust therefore Mutual Information (MI) algorithms for keywords extraction were considered: MI algorithm based on stance class (MISC), MI algorithm based on customized class (MICC), Pointwise Mutual Information Algorithm (PMI).
Mutual information is a symmetric, a non-negative measure of the common information in two variables. It has often been used for clustering words, in which scenario, MI can be interpreted as the amount of information the presence and absence of a term contribute to making the correct classification on a class. The difference between MICC

and MISC in that in the first case the title and body for refuting/discuss words used separately as different documents and for second their concatenated. PMI approach allowed extract keywords when known the refuting/discuss words. Usually, this algorithm shows high performance on large datasets, since in the case under consideration a small dataset with some classes extremely poor, the PMI method is not considered.

The third type of feature that was used as input of the model is vectorized implementation of text also known as word embedding models. Embeddings creation is a crucial task in any modern language-related machine learning problem. We can divide that task into two large families, namely word embeddings and sentence embeddings models.

The original idea for first family was proposed in (Mikolov et al., 2013). Several other successful approaches was based on that work, such as: GloVe (Pennington et al., 2014), FastText (Bojanowski et al., 2016; Joulin et al., 2016), ELMo (Peters et al., 2018).
GloVe model implements the combination of two large families of the models for learning word vectors: 1) global matrix factorization methods (latent semantic analysis (LSA), e.t.c) 2) local context window methods (skip-gram model) That approach eliminates the drawbacks of each family. Global matrix factorization makes effective usage of statistics in the corpus but misses word analogies. While local context window methods show great performance on analogy task but miss statistical information.
FastText model implements the idea of tacking into account morphology to get a piece of additional information about the meaning of words. This idea comes from skip-gram model adaptation to morphologically rich languages, such as Turkish or Finnish.
Embeddings from Language Models (ELMo) representations is the state of the art algorithm for word embeddings. Its algorithm uses deep learning approaches. The main idea is to use vectors derived from a bidirectional LSTM that is trained with a coupled language model objective on a large text corpus. Unlike most previously used word embeddings, ELMo word representations are functions of the entire input sentence.

Second family baseline are already well known bag of words approach. As you understand such approaches also could be implemented for FNC task. The good examples of modern sentence embeddings are: Skip Thoughts (Kiros et al., 2015), DiscSent (Jernite et al., 2017), Quick thoughts (Logeswaran and Lee, 2018).

Any approach that was listed above can be used for text representation. However the baseline approach uses the word level description, so first family model outputs look more promising as features for input vector concatenations, cause they encode similar parameters. In original work, only word2vec format was implemented.
For document comparison, several approaches were used. Firstly we can make an average output vector for two documents, and after compute cosine similarities between them. The calculated distance will be an additional feature that the author wanted to get. Cosine similarity is a straight forward, easily implemented algorithm, however, some valuable information will be lost during the vector averaging process.
So second approach for similarity measurement was implemented, namely World Mover's Distance (WMD). It calculates the minimum cumulative distance that all word vectors from document 1 need to travel to match document 2. The travel distance should match between the two sets of words in the two documents.

3.5. Ensemble

After understanding the promising features. Author uses them as inputs for simple neural network proposed by UCL team. And after inspects accuracy of all of the six individual models. The results of that comparison are shown in the table below.
The 10-fold validation reveals that all models perform similarly in classifying category discuss and unrelated, while vary in category agree and disagree. Especially, the results of agree and disagree category recall from the previous table are presented with visualized bar charts, figure 3.5..

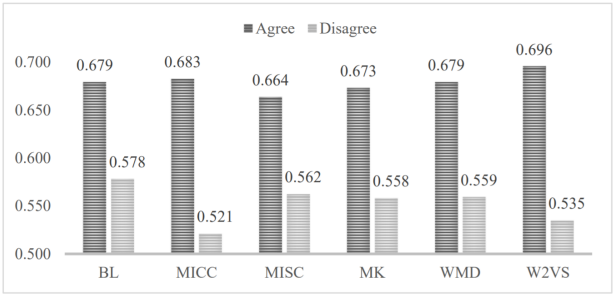

Figure 4: 10-fold average agree & disagree recall rate of different models

For word embedding approach, WMD algorithm is expensive in computation, but the the extra cost in training time does not bring a significant improvement in the related category. Word2vec shows significant improvement for agree category, however, the validation also reveals its weakness in disagree. Keywords based algorithms perform mediocre in all perspectives. The two Mutual information algorithms generate a very similar outcome, and between them, MICC has a slight advancement in the classifying category agree, which is better than baseline. Considering MICC achieved a higher grade, it was chosen for the ensemble learning, while MISC was not.
Based on chosen in single NN analysis was performed 2 ensemble comparisons. In this propose was consider 2 voting mechanisms: summation and concatenation rule. The difference is that argmax function in the ends of rules applies to summated or concatenated (increasing in n times) prediction vectors. As describe above from single model analysis was chosen such models: Baseline, MICC, MK, W2VS, WMD. The baseline model has shown an overall good performance in all categories in the 10-fold cross-validation, what's more, it is cheap in

	Agree	Disagree	Discuss	Unrealted	Recall	Grade	Loss
Baseline	0.679	**0.578**	0.882	0.991	0.941	90.699	**16548**
MICC	0.683	0.521	0.884	0.991	0.941	90.606	16785
MISC	0.664	0.562	**0.885**	0.990	0.940	90.516	16811
Manually Keywords	0.673	0.558	0.882	**0.992**	0.941	90.536	16613
WMD	0.679	0.559	0.883	0.991	0.941	90.674	16557
Word2Vec	**0.696**	0.535	0.881	0.991	**0.942**	**90.771**	16568

Table 2: Models evaluation.

computation and its training time is short, therefore, the baseline model was considered as the major model in the ensemble learning. Different models were combined with the baseline models following two combining rules: concatenation and summation rule. An ensemble of two models went through 10-fold cross-validation firstly, next, the simplest three single neural network models, baseline, MK and MICC were selected for three model ensemble validation.

A comparison of FNC-1 grade for different ensemble configurations with two combining algorithms is visualized in Figure 3.5.. The summation rule showed the best performance for each case of considered ensemble models.

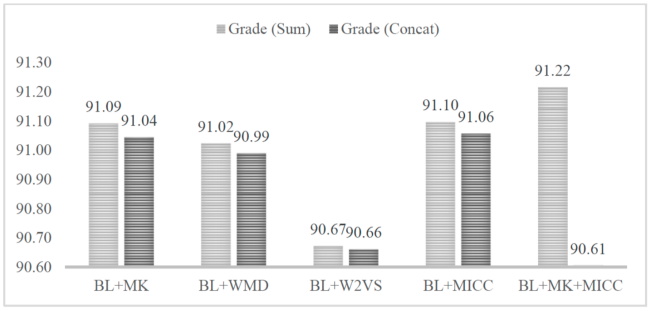

Figure 5: 10-fold cross validation for Ensemble learning models.

As shown in figure 3.5. author chosen 2 best pairs of ensemble models and then concatenate it to an ensemble of 3 models, which as expected get the best results for each class with summation rule.

4. Analysis

As was described in 3.3 used dataset had very unbalanced classes of data: unrelated class more than 43 times bigger than disagree. Also as consider by recall metrics result we can see a strong correlation between metric value and size of the presented class, figure 4..

As shown in figure 4. dependence of the value of the recall metric for the class on the amount of data of this class in the dataset we can notice that for the first 3 values there is almost a direct relationship, which may indicate insufficient data size for the model. And for the unrelated class, saturation occurs because with an increase in the amount of data the value of the metric increases less and less. Accordingly, we can say that only one class has enough data.

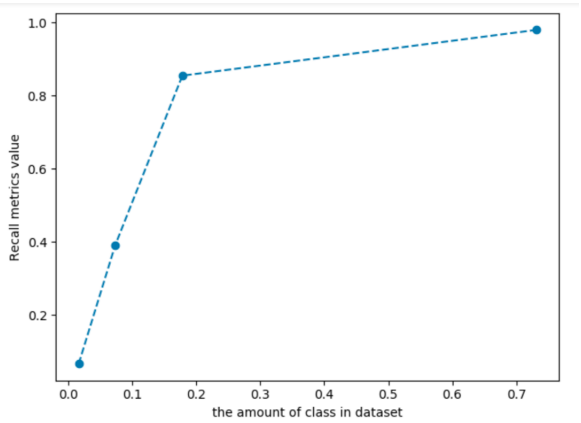

Figure 6: The relationship between class size and recall metrics

Also, the paper was not considered the weighted voting of ensemble models. To increase the value of recall metric in most weak classes we created weights matrix for each of three ensemble models:

$$M_1 = diag(1, 1.154, 1.042, 1.003)$$
$$M_2 = diag(1.023, 1.415, 1.041, 1)$$
$$M_3 = diag(1.253, 1, 1, 1.010)$$

where M_1 weights for baseline model, M_2 for MK and M_3 for MICC. Coefficients was calculated based on recalls value for each classes of each models. It's allow us to get much better results in Table 3.

	Ens	WE	Athene	UCL	Talos
Agree	0.391	0.405	0.447	0.440	**0.585**
Disagree	0.067	0.078	**0.095**	0.066	0.019
Discuss	**0.855**	0.851	0.809	0.814	0.762
Unrelated	0.98	0.979	**0.992**	0.979	0.987
Grade	82.32	**82.39**	81.97	82.72	82.02

Table 3: Models evaluation.

As we can see from Table 3,where Ens - Ensemble model, WE - Weighted Ensemble model, applying weights allowed to increase recall metrics in more weak classes and increase the main grade as a top.

In consider, work author pays great attention to the ease and speed of algorithms despite the fact that the goal of the problem is accuracy. In this context, when analyzing individual and paired models, it is possible to compare models not only with baseline but with a combination of other models, using different class weights to obtain a more complete picture of

the analysis.

The word2vec performance in the author's pipeline showed unexpectedly low results. The first remark about that issue is that this is the consequence of usage old model, in the methodology we listed the more promising modern approaches.

However, the second remark is that the author removed the most frequent language words before applying of word2vec model. However, even in the original approach, that problem was discussed and a special approach was introduced to counter the class disbalance. It means that you can use word2vec even without eliminating frequent words. So there is a possibility that on the full-text document this approach would give a far better result.

The third remark is the structure of the author's neural network. It has a shallow structure that strongly boosts the speed of computations. However Google news word2vec pre-trained model encodes a huge amount of information about a single word. What leads to the assumption that the author's neural network simply lacks the classification power to use the distance between vectors as input.

5. Conclusion

In this work, the author managed to achieve better results in comparison with the board of the competition leaders. However, as was noted in the analysis section, an additional comparison can be made with the improvement of some points:

- applying the data balancing algorithm for training
- consider other combinations of not-so-lightweight models,
- apply proportional balancing in voting during analysis
- apply the considered modern encoding approaches words.

However, despite these drawbacks, we managed to slightly improve the result obtained in the work by adding weights to the best model. This result shows the viability of voting several models of their best qualities. But as discussed above, this area requires additional analysis.

6. Bibliographical References

Bojanowski, P., Grave, E., Joulin, A., and Mikolov, T. (2016). Enriching word vectors with subword information. *CoRR*, abs/1607.04606.

Chen, J. (2017). Rnns for stance detection between news articles.

Cisco-Talos. (2018). Cisco-talos/fnc-1, Jun.

Fajcik, M., Burget, L., and Smrz, P. (2019). BUT-FIT at semeval-2019 task 7: Determining the rumour stance with pre-trained deep bidirectional transformers. *CoRR*, abs/1902.10126.

Hanselowski.). hanselowski/athene$_system$.

Jernite, Y., Bowman, S. R., and Sontag, D. A. (2017). Discourse-based objectives for fast unsupervised sentence representation learning. *CoRR*, abs/1705.00557.

Joulin, A., Grave, E., Bojanowski, P., and Mikolov, T. (2016). Bag of tricks for efficient text classification. *CoRR*, abs/1607.01759.

Kiros, R., Zhu, Y., Salakhutdinov, R., Zemel, R. S., Torralba, A., Urtasun, R., and Fidler, S. (2015). Skip-thought vectors. *CoRR*, abs/1506.06726.

Leskovec, J., Rajaraman, A., and Ullman, J. D. (2016). Mining of massive datasets. Cambridge University Press.

Liao, W. (2018). Stance Detection in Fake News: An Approach based on Deep Ensemble Learning. Ph.D. thesis, 08.

Logeswaran, L. and Lee, H. (2018). An efficient framework for learning sentence representations. *CoRR*, abs/1803.02893.

Mikolov, T., Chen, K., Corrado, G., and Dean, J. (2013). Efficient estimation of word representations in vector space.

Pennington, J., Socher, R., and Manning, C. (2014). Glove: Global vectors for word representation. volume 14, pages 1532–1543, 01.

Peters, M. E., Neumann, M., Iyyer, M., Gardner, M., Clark, C., Lee, K., and Zettlemoyer, L. (2018). Deep contextualized word representations. *CoRR*, abs/1802.05365.

Rakholia, N. (2017). Is it true? deep learning for stance detection in news.

Rao, P. (2019). Transfer learning in nlp for tweet stance classification, Aug.

Uclnlp. (2018). uclnlp/fakenewschallenge, May.

Xu, B., Mohtarami, M., and Glass, J. R. (2019). Adversarial domain adaptation for stance detection. *CoRR*, abs/1902.02401.